

CLAIMS

1-13. (Canceled)

14. (Previously Presented) A computer implemented method of managing access to a storage resource for one of a plurality of network-based applications in a multiple server storage system, the method comprising:

obtaining a resource identifier from a front end server;

utilizing the resource identifier to lookup, in a resource lookup store of a lookup partitioning service server, a partition of a storage server associated with the resource identifier;

in an event the partition of the storage server is associated with the resource identifier, granting access to the storage resource by providing a location of the partition of the storage server to the front end server; and

in an event no partition of the storage server is associated with the resource identifier:

failing to locate a mapping to the storage resource;

determining a load balancing factor for each storage partition of a plurality of storage partitions;

using the load balancing factors to determine a new storage partition in which a new storage resource should be created and creating the new storage resource in the new storage partition;

mapping the resource identifier to the new storage partition in the resource

lookup store;

and

providing a location of the new storage partition to the front end server,

wherein determining a load balancing factor for each storage partition comprises:

given n storage partitions, determining the number of mapping counts C for each of the n storage partitions; and

calculating a load balancing factor (LBF_m) for each storage partition m using the following:

$$LBF_m = (1/C_m) / (1/C_1 + 1/C_2 + \dots + 1/C_n).$$

15. (Previously Presented) The method of Claim 14, wherein the location of the partition of the storage server is on one of a plurality of storage servers.

16. (Previously Presented) The method of Claim 14, wherein the plurality of storage partitions include a primary storage partition and a redundant storage partition each containing the storage resource.

17. (Previously Presented) The method of Claim 16, wherein the primary storage partition and the redundant storage partition are each located on separate storage servers of a plurality of storage servers.

18. (Previously Presented) The method of Claim 17, wherein, if the primary storage partition is unavailable, the storage server location is the redundant storage

partition.

19. (Currently Amended) The method of Claim 14, further comprising determining which lookup partitioning service server of a plurality of lookup partitioning service servers will provide wherein the looked-up storage server location in response to the resource identifier.

20. (Previously Presented) The method of Claim 19, wherein determining which lookup partitioning service server will provide the looked-up storage server location comprises processing the resource identifier through a hash function to provide a hashed resource identifier associated with a particular lookup partitioning service server.

21. (Previously Presented) The method of Claim 20, wherein each lookup partitioning service server is associated with a predetermined set of hashed resource identifiers.

22. (Previously Presented) The method of Claim 14, further comprising moving the storage resource from one storage partition to another storage partition and updating the resource lookup store with the another storage partition.

23. (Canceled)

24. (Canceled)

25. (Currently Amended) The method of Claim 14, wherein the load balancing factors are further based on a value selected from the values consisting of: a mapping number and a manual weighting value.

26. (Currently Amended) The method of Claim 14 further comprising adjusting a manual weighting value to increase a usage of ~~[[of]]~~ a storage server.

27. (Currently Amended) The method of Claim 14, further comprising adjusting a manual weighting value to decrease a usage of ~~[[of]]~~ a storage server.

28. (Previously Presented) A computer readable storage medium containing computer-executable instructions for performing a method of managing access to a storage resource for one of a plurality of network-based applications in a multiple server storage system, the computer-executable instructions comprising instructions comprising:

receiving a resource identifier associated with the storage resource from a front end server;

utilizing the resource identifier to lookup, in a resource lookup store of a lookup partitioning service server, a storage partition associated with the resource identifier;

in an event the storage partition is associated with the resource identifier:

locating the storage partition of a storage server; and

sending the location of the storage partition to the front end server to grant access to the storage resource; and

in an event the storage partition is not associated with the resource identifier:

 failing to locate a mapping to the storage resource;

 determining a load balancing factor for each storage partition of a plurality of storage partitions;

 using the load balancing factors to determine a new storage partition in which a new storage resource should be created and creating the new storage resource in the new storage partition;

 mapping the resource identifier to the new storage partition in the resource lookup store; and

 sending a location of the new storage partition to the front end server, wherein determining a load balancing factor for each storage partition comprises:

 given n storage partitions, determining the number of mapping counts C for each of the n storage partitions; and

 calculating a load balancing factor (LBF_m) for each storage partition m using the following:

$$LBF_m = (1/C_m) / (1/C_1 + 1/C_2 + \dots + 1/C_n).$$

29. (Previously Presented) The computer-readable storage medium of Claim 28, further comprising determining which lookup partitioning service server of a plurality of lookup partitioning service servers will locate the storage partition in response to the resource identifier.

30. (Previously Presented) The computer-readable storage medium of Claim 28, further comprising relocating the storage resource from one storage partition to a different storage partition and updating the mapping of the resource identifier at the lookup partitioning service server.

31. (Canceled)

32. (Canceled)

33. (Previously Presented) The computer-readable storage medium of Claim 28, wherein the load balancing factors are further based on a value selected from the values consisting of: a mapping number and a manual weighting value.

34-49. (Canceled)

50. (Previously Presented) A lookup partitioning server comprising:

- a processing unit;
- at least one primary lookup partition; and
- at least two redundant lookup partitions which mirror two respective different primary lookup partitions stored on other look-up partitioning servers;
- a memory configured to store computer-executable instructions configured to manage access to a plurality of storage resources at a plurality of storage servers, the computer-executable instructions performing acts comprising:

receiving a resource identifier associated with a storage resource from a front end server;

utilizing the resource identifier to lookup, in a resource lookup store, a storage partition associated with the resource identifier;

in an event a particular storage partition is associated with the resource identifier:

locating the particular storage partition of a storage server; and

sending a location of the particular storage partition of the storage server to the front end server to grant access to the storage resource; and

in an event no particular storage partition is associated with the resource identifier:

failing to locate a mapping to the storage resource;

determining a load balancing factor for each storage partition of a plurality of storage partitions;

using the load balancing factors to determine a new storage partition in which a new storage resource should be created and creating the new storage resource in the new storage partition;

mapping the resource identifier to the new storage partition in the resource lookup store; and

sending a location of the new storage partition to the front end server,

wherein determining a load balancing factor for each storage partition comprises:

given n storage partitions, determining the number of mapping counts C for each of the n storage partitions; and
calculating a load balancing factor (LBF_m) for each storage partition m using the following:

$$LBF_m = (1/C_m) / (1/C_1 + 1/C_2 + \dots + 1/C_n).$$

51. (Canceled)

52. (Currently Amended) The lookup partitioning server of Claim 50, wherein using the load balancing factors to determine a new storage partition comprises:

generating a random number R , where R is a real number between 0 and 1;

ranking the load balancing factors in ascending order;

locating [[a]] another particular storage partition, where the sum of all the load balancing factors ranked lower than the load balancing factor corresponding to the another particular storage partition is less than or equal to R , and the sum of all the load balancing factors ranked lower than the load balancing factor corresponding to the another particular storage partition plus the load balancing factor corresponding to the particular storage partition is greater than R .

53. (Previously Presented) The method of Claim 14, wherein using the load balancing factors to determine a new storage partition comprises:

generating a random number R , where R is a real number between 0 and 1;

ranking the load balancing factors in ascending order;

locating a particular storage partition, where the sum of all the load balancing factors ranked lower than the load balancing factor corresponding to the particular storage partition is less than or equal to R , and the sum of all the load balancing factors ranked lower than the load balancing factor corresponding to the particular storage partition plus the load balancing factor corresponding to the particular storage partition is greater than R .

54. (Previously Presented) The computer-readable storage medium of Claim 28, wherein using the load balancing factors to determine a new storage partition comprises:

generating a random number R , where R is a real number between 0 and 1;

ranking the load balancing factors in ascending order;

locating a particular storage partition, where the sum of all the load balancing factors ranked lower than the load balancing factor corresponding to the particular storage partition is less than or equal to R , and the sum of all the load balancing factors ranked lower than the load balancing factor corresponding to the particular storage partition plus the load balancing factor corresponding to the particular storage partition is greater than R .